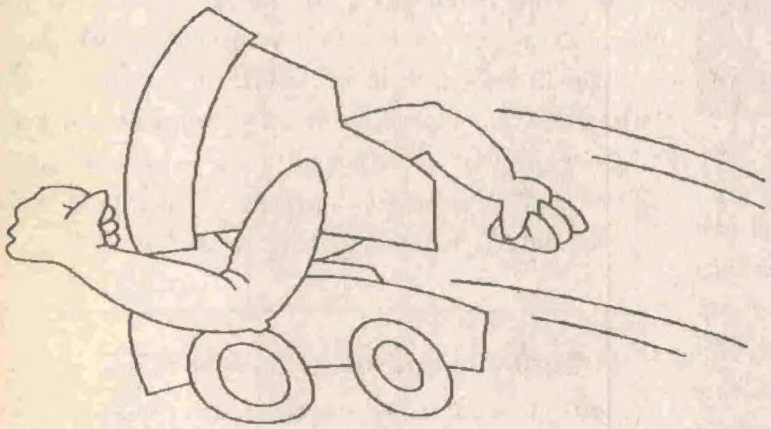
ชัยวิทย์ พู่วณิชย์ และ สุทัศน์ แซ่ตั้ง

Java ภาษาคอมพิวเตอร์แห่งอนาคต



เพิ่มความเข้าใจเกี่ยวกับภาษา Java ซึ่งสามารถพัฒนาครั้งเดียว และนำไปใช้ได้ในหลาย ๆ แพลตฟอร์ม

หลาย ๆ คนคงรู้จัก Java กันมาบ้างแล้ว สำหรับอีกหลาย ๆ คนที่ยังไม่รู้จัก บทความนี้คงช่วยเพิ่มความเข้าใจเกี่ยวกับ Java ได้บ้าง แต่ก่อนที่เราจะมาคุยกันถึง Java ที่ไม่ได้หมายถึงกาแฟหรือเกาะชวา แต่หมายถึงภาษาคอมพิวเตอร์ที่กำลังบูมอยู่ในขณะนี้กัน เราควรจะรู้เกี่ยวกับเรื่องการเขียนโปรแกรมเชิงวัตถุ (object-oriented programming) กันก่อนเพื่อให้เข้าใจ Java ได้ดีขึ้น จากนั้นเราจะมาคุยกันเกี่ยวกับประวัติและแนวคิดของภาษา Java สุดท้ายจะคุยกันถึงการเขียนโปรแกรมด้วย Java และเปรียบเทียบ Java กับ C++ ซึ่งเป็นภาษาที่สนับสนุนแนวคิดในการเขียนโปรแกรมเชิงวัตถุเช่นเดียวกัน รวมถึง library และ package ของภาษา java ที่ Sun ให้มากับตัว JDK ว่ามีอะไรกันบ้าง

การเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming)

เมื่อพูดถึงการเขียนโปรแกรมเชิงวัตถุแล้ว จะมีลักษณะแตกต่างกับการเขียนโปรแกรมแบบโครงสร้าง (Structured Programming) อยู่มากทีเดียว โดยการเขียนโปรแกรมเชิงวัตถุจะมองทุกอย่างเป็นวัตถุ (object) และแต่ละวัตถุมีการติดต่อซึ่งกันและกันทางช่องที่เปิดให้ (methods/member functions) หรือมองเป็นแบบคำนาม แต่การเขียนโปรแกรมแบบโครงสร้างนั้นจะมองทุกอย่างเป็นขั้นตอนการทำงาน (procedure) มีข้อมูลเข้า (input data) ส่งให้แก่ขั้นตอนการทำงานนั้น ๆ แล้วประมวลผลส่งเป็นข้อมูลออก (output data) มาให้ หรือมองเป็นแบบกริยาที่กระทำกับข้อมูล ตัวอย่างเช่น การเขียนโปรแกรมเกี่ยวกับแผนกจัดซื้อ ถ้าเป็นการเขียนโปรแกรมแบบโครงสร้าง เราก็จะมองว่าในการจัดซื้อแต่ละครั้งจะมีขั้นตอนการทำงานอะไรบ้าง ซึ่งน่าจะประกอบด้วย การทำใบจัดซื้อ การอนุมัติใบจัดซื้อ การพิมพ์ใบจัดซื้อ เป็นต้น แต่ในขณะที่การเขียนโปรแกรมเชิงวัตถุจะมองว่าในการจัดซื้อประกอบด้วยใบจัดซื้อ ในใบจัดซื้อประกอบด้วยผู้ขาย สินค้าแต่ละรายการ เป็นต้น

นอกจากนี้แล้ว ในการเขียนโปรแกรมเชิงวัตถุควรจะต้องทำความเข้าใจถึงคำศัพท์พื้นฐาน บางคำซึ่งเกี่ยวกับแนวคิดเชิงวัตถุ (object-oriented concepts) อีกด้วย นั่นคือ

1. Object คือ วัตถุที่รวมข้อมูล และคำสั่งที่จะดำเนินการกับข้อมูลนั้น ๆ เข้าไว้เป็นหน่วยเดียวกัน

2. Class คือ การกำหนดคุณลักษณะของข้อมูลและคำสั่งในวัตถุ หรืออาจจะกล่าวได้ว่าคลาสเป็นพิมพ์หรือเทมเพลตของวัตถุก็ได้ เช่น เราสร้างคลาสชื่อจอภาพ ซึ่งมีคุณลักษณะว่าจอภาพจะต้องมีหลอดภาพ ปุ่มเปิดปิด และฐานสำหรับตั้งจอภาพ เป็นต้น ดังนั้นเมื่อเราสร้างวัตถุ a ให้เป็นสมาชิกของคลาสจอภาพ เราก็จะได้วัตถุ a ที่มีคุณลักษณะตามคลาสจอภาพ คือ มีหลอดภาพ ปุ่มเปิดปิด และฐานสำหรับตั้งจอภาพ เป็นต้น

3. Encapsulation คือ การซ่อนรายละเอียดของข้อมูลไว้ การเข้าถึงข้อมูลในแต่ละวัตถุนั้นจะเข้าถึงผ่านจากคำสั่ง หรือฟังก์ชันที่เรากำหนดไว้เท่านั้น ทำให้ข้อมูลไม่ถูกเข้าถึง หรือเปลี่ยนแปลงจากภายนอก การห่อหุ้มข้อมูล (data encapsulation) หรือการซ่อนข้อมูล (data hiding) นี้เป็นจุดเด่นของการเขียนโปรแกรมเชิงวัตถุ

4. Inheritance คือ การสืบทอดซึ่งเป็นจุดเด่นอีกประการหนึ่งของการเขียนโปรแกรมเชิงวัตถุ ทำให้เราสามารถนำสิ่งที่มีอยู่แล้วกลับมาใช้ใหม่ได้ (reusability) หรือนำมาเพิ่มคุณลักษณะต่าง ๆ เข้าไปเพื่อให้มีประสิทธิภาพมากขึ้น เช่น จากตัวอย่างที่แล้วถ้าเรามีคลาสชื่อแป้นพิมพ์ และคลาสชื่อ CPU เพิ่มขึ้นมา เราสามารถสร้างคลาสใหม่ที่การสืบทอดคุณลักษณะของคลาสทั้งสามได้ โดยอาจจะตั้งชื่อคลาสที่จะสร้างขึ้นมานี้ว่าคอมพิวเตอร์ซึ่งมีคุณลักษณะคือมีจอภาพ มีแป้นพิมพ์ มี CPU เป็นต้น

5. Polymorphism คือ การที่เรากระทำหรือใช้คำสั่งเหมือนกันกับประเภทของวัตถุที่ต่างกัน ซึ่งจะทำให้ได้ผลที่ต่างกันด้วย เช่น ถ้าเรานำตัวเลข 2 ตัวมาบวกกัน กับการบวกเวลา 2 ชุดเข้าด้วยกัน จะเป็นการบวกที่ต่างกัน เพราะการบวกตัวเลขใช้การบวกปกติ ในขณะที่การบวกเวลาไม่ได้เป็นการบวกปกติ คือในหลักวินาที และนาทีจะบวกด้วยฐาน 60 ในขณะที่หลักชั่วโมงจะบวกด้วยฐาน 24 แต่ในการเขียนโปรแกรมเชิงวัตถุเราสามารถเขียนรูปแบบคำสั่งเหมือนกันคือ no1 + no2, time1 + time2 แต่ในการทำงานจะเรียกใช้การบวกที่ต่างกันได้โดยขึ้นกับชนิดของวัตถุที่เรียกใช้คำสั่ง เป็นต้น

เข้าใจภาษา Java

เมื่อทราบแนวคิดพื้นฐานของการเขียนโปรแกรมเชิงวัตถุแล้ว ต่อไปเราจะคุยกันเกี่ยวกับภาษา Java ซึ่งเป็นภาษาในการเขียนโปรแกรมเชิงวัตถุภาษาหนึ่งที่นิยมใช้ในปัจจุบัน ขอเริ่มต้นด้วยประวัติของ Java สักนิดก่อน

โดยแนวคิดของบริษัท Sum Microsystem ที่ต้องการจะพัฒนาตัวจัดการอุปกรณ์อิเล็กทรอนิกส์ให้สามารถทำงานได้โดยไม่ขึ้นกับแพลตฟอร์ม (Platform) ไม่ขึ้นกับชนิดของไมโครโปรเซสเซอร์ หรือฮาร์ดแวร์ ทำให้ Java ถือกำเนิดในปี พ.ศ. 2534 (ค.ศ. 1991) ซึ่งขณะนั้นใช้ชื่อว่า Oak หลังจากนั้นในปี พ.ศ. 2536 (ค.ศ. 1993) อินเทอร์เน็ตและเว็บได้กลายเป็นที่นิยมกันมาก ทาง Sun พยายามพัฒนาให้ Oak กลายมาเป็นภาษาโปรแกรมที่มีแพลตฟอร์มเป็นกลาง และเป็นภาษาสำหรับพัฒนาโปรแกรมบนเว็บได้ด้วย และได้เปลี่ยนชื่อจาก Oak มาเป็น Java ในที่สุด

Sun ออก Java Development Kit (JDK) ซึ่งเป็น API สำหรับผู้สนใจเขียนโปรแกรม Java นำไปใช้พัฒนา โดยสามารถ download ได้ฟรี ปัจจุบันเป็นเวอร์ชัน 1.1 และคาดว่าเวอร์ชัน 1.2 คงจะออกในต้นปี พ.ศ. 2542 (ค.ศ. 1999) ปัจจุบันมีเครื่องมือพัฒนาภาษา Java ที่เป็น visual ให้เลือกมากมาย อาทิเช่น Jbuilder ของ Borland, Java Studio ของ Sun, Visual Age for Java ของ IBM, JDesignerPro ของ BulletProof และอื่น ๆ อีกมากมาย

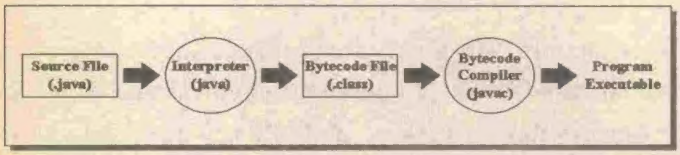
Java มีคุณสมบัติที่เด่นหลายประการ คุณสมบัติประการหนึ่งก็คือสามารถทำงานได้โดยไม่ขึ้นกับแพลตฟอร์ม เราทราบกันแล้วว่า Java ได้ถูกออกแบบมาให้สามารถทำงานได้บนระบบคอมพิวเตอร์หลาย ๆ ประเภท โดยทั่วไปโปรแกรมที่จะทำงานบนคอมพิวเตอร์จะต้องถูกแปลจากภาษาระดับสูง เช่น C, Pascal ไปเป็นภาษาเครื่องโดยมีตัวแปลภาษาทำหน้าที่ในการแปลงภาษาระดับสูงไปเป็นภาษาเครื่อง คอมพิวเตอร์แต่ละประเภทจะมีตัวแปลภาษาทำหน้าที่ในการแปลงภาษาที่แตกต่างไม่เหมือนกัน เป็นเหตุให้ซอฟต์แวร์ตัวนั้นมีหลายเวอร์ชัน ซึ่งแต่ละเวอร์ชันก็สามารถทำงานได้บนคอมพิวเตอร์ประเภทนั้น ๆ เท่านั้น แต่ลักษณะสถาปัตยกรรมของ Java เป็นแบบ binary ซึ่งเครื่องคอมพิวเตอร์ทุกประเภทสามารถทำการประมวลผลแบบ binary code ได้ จึงส่งผลให้ Java ทำงานได้บนคอมพิวเตอร์หลายแพลตฟอร์ม และอีกเหตุผลหนึ่งเป็นเพราะ Java ใช้มาตรฐานข้อมูลแบบ IEEE และ Java มี library สำหรับรูทีน (Routine) ต่าง ๆ ช่วยในการประมวลผลข้อมูลภายในระบบปฏิบัติต่าง ๆ library เหล่านี้จะถูกเรียกใช้เมื่อโปรแกรมทำงานภายใต้ระบบปฏิบัติที่ Java รองรับ

นอกจากนั้น Java ได้สร้างข้อมูลชนิดชุดตัวอักษร (string) โดยใช้มาตรฐานของ IEEE ซึ่งทำให้สามารถทำงานร่วมกับระบบที่มีการใช้มาตรฐานของ IEEE ได้ด้วย ข้อมูลชนิดชุดตัวอักษรช่วยให้สามารถทำงานกับข้อมูลตัวอักษรประเภทต่าง ๆ ได้ง่าย ชุดตัวอักษรใน Java ไม่ใช่อะเรย์ของตัวอักษรเหมือนในภาษา C แต่จะมีลักษณะเหมือนข้อมูลชนิดชุดตัวอักษรในภาษา basic มากกว่า

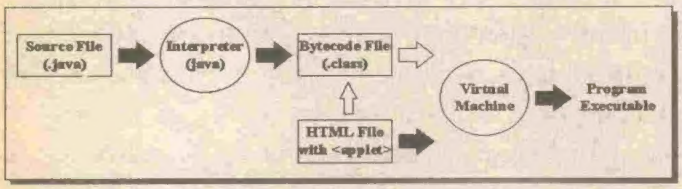
อีกคุณสมบัติหนึ่งที่น่าสนใจของ Java คือ เราไม่ต้องยุ่งกับการจัดการหน่วยความจำ Java จะจัดการหน่วยความจำให้โดยอัตโนมัติ Java ไม่มีตัวแปรชนิด pointer ทำให้เป็นภาษาที่ความปลอดภัย (secure) และแข็งแรง (robust) เพราะจะไม่มีการอ้างอิงนอกเขต ซึ่งเป็นสาเหตุหนึ่งในการทำให้เครื่อง crash และ Java เองยังมีคุณสมบัติที่เรียกว่าการรวบรวมขยะ (garbage collection) ซึ่งเป็นการยกเลิกการจองตำแหน่งในหน่วยความจำที่มีการจองแต่ไม่มีการงานเป็นเวลานาน เพื่อให้สามารถใช้งานหน่วยความจำนั้นได้อย่างมีประสิทธิภาพ

ใน Java เราสามารถเขียนโปรแกรมเล็ก ๆ หรืออาจจะเรียกว่าเป็นฟังก์ชันเล็ก ๆ ที่สามารถทำงานกับเว็บได้โดยฝังไว้ใน HTML ของเรา ซึ่งเรียกว่า java applet ความแตกต่างระหว่าง java applet กับ java application ปกติคือใน java application ปกติจะมีฟังก์ชัน main แต่ใน java applet จะไม่มีฟังก์ชัน main, java applet ต้อง derive มาจากคลาส java.applet.Applet และจะทำงานจากการฝังตัวเองไว้กับ HTML ส่วน java application สามารถทำงานได้ด้วยตัวมันเอง ดังที่เห็นในรูปที่ 1 และ 2 ตามลำดับ

อ้อ... อย่าเข้าใจผิดกับ java script ของ Netscape นะครับ java script เป็นเพียง script file (คล้าย ๆ กับ batch file ในดอส) ที่ผสมอยู่ใน HTML เพื่อเพิ่มประสิทธิภาพในการทำงานของ HTML เท่านั้น ซึ่งรูปร่างหน้าตาของ script ที่เขียนอาจจะคล้ายกับ Java ของ Sun จึงอาจจะทำให้เกิดความสับสนได้



รูปที่ 1 ขั้นตอนการทำงานของ compiler และการเรียกใช้ของ java application



รูปที่ 2 ขั้นตอนการทำงานของ compiler และการเรียกใช้ของ java applet

การที่ java applet ถูกออกแบบมาทำงานที่ local โดย load จาก host ความปลอดภัยจึงเป็นสิ่งสำคัญ และด้วยเหตุนี้ java applet จึงถูกกำหนดให้มีความสามารถน้อยกว่า java application ซึ่งตามข้อกำหนดของ Sun กำหนดให้ java applet ไม่สามารถทำสิ่งต่าง ๆ ดังนี้ได้

1. ไม่สามารถเรียกโปรแกรมที่ local ใด ๆ ทำงานได้

2. ไม่สามารถติดต่อกับ host อื่น ๆ นอกจาก host ที่เป็นเจ้าของมัน (origination host) เท่านั้น

3. ไม่สามารถอ่าน หรือเขียน system file ที่ local คอมพิวเตอร์ได้ (ใช้ได้เฉพาะกับ browser ของ netscape เท่านั้น ในส่วนของ brwoser อื่น ๆ อาจจะสามารถทำได้เพราะในส่วนนี้ Sun ไม่ได้กำหนดไว้เป็นมาตรฐานตายตัว)

4. ไม่สามารถค้นหาข้อมูลอื่น ๆ เกี่ยวกับ local คอมพิวเตอร์นั้น ๆ ได้นอกจากเวอร์ชันของ Java ที่ใช้ ชื่อ และเวอร์ชันของระบบปฏิบัติการที่ใช้ และตัวอักษรที่ใช้ในการแบ่งแยกไฟล์ (\ หรือ /) path (: หรือ ;) ขึ้นบรรทัดใหม่ (\n หรือ \r\n) โดยเฉพาะอย่างยิ่งไม่สามารถค้นหาชื่อและ e-mail address ของผู้ใช้ได้

การเขียนโปรแกรม Java เบื้องต้น

ในด้านไวยากรณ์ของภาษาแล้ว Java มีความคล้ายกับ C++ อยู่มาก แต่ก็มีส่วนไม่เหมือนกับ C++ ด้วยเช่นกัน ขอคุยถึงส่วนคล้ายกับ C++ กันก่อนก็แล้วก่อน ในแง่ของ data type ที่ Java มีให้ทั้งหมด 8 แบบ คือ byte (8 bit), short (2 bit), int (4 bit), long (8 bit), float (4 bit), double (8 bit) boolean (true/false) ในส่วนของ data type แบบ char นั้น Java ใช้ 16 bit แบบ unicode (unicode เป็นเซตของตัวอักษรขนาด 16 bit ซึ่งประกอบไปด้วยสัญลักษณ์ ตัวอักษรของภาษาต่าง ๆ ทั่วโลก และ ASCII ก็เป็นเซตย่อย (subset) ของ unicode ด้วย) ในส่วนการกำหนดตัวแปร การกำหนดค่า ค่าคงที่ อะเรย์ ตัวดำเนินการต่าง ๆ รวมถึงคำสั่งควบคุมต่าง ๆ จะเหมือนกับ C++

ตารางที่ 1 แสดงการเปรียบเทียบความสามารถระหว่าง java applet กับ java application ในการทำงานแบบต่าง ๆ

ความสามารถที่ทำได้	Remote Applet	Local Applet	Applet Viewer	Application
อ่าน local file	✗	✗	✓	✓
เขียน local file	✗	✗	✓	✓
ค้นหาข้อมูลอื่น ๆ ของคอมพิวเตอร์	✗	✗	✓	✓
ลบ file	✗	✗	✗	✓
เรียกโปรแกรมอื่นทำงาน	✗	✗	✓	✓
อ่าน user.name	✗	✓	✓	✓
ติดต่อ network port ของ host	✓	✓	✓	✓
ติดต่อ network port ของ host อื่น	✗	✓	✓	✓
load java library	✗	✓	✓	✓
เรียกคำสั่ง exit	✗	✗	✓	✓
สร้าง pop-up window	✓	✓	✓	✓

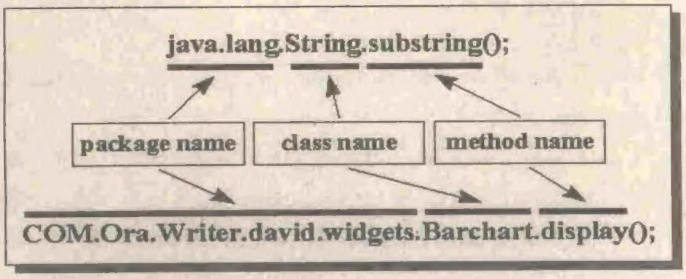
หมายเหตุ : -
Remote applet = Load โดยใช้ URL
Local applet = Load แบบ local file
Applet viewer = Load ด้วย applet viewer
Application = Java application

ในกรณีของ C++ เราใช้คำสั่ง #include ในเข้าถึง library file แต่ใน Java เราจะใช้ import ในการเข้าถึง library ของ Java แต่จะต่างกันที่คำสั่ง #include ของ C++ จะทำการ load header file ซึ่งเป็น text file ของ C++ (.h, .hpp) รวมกับ source code ก่อนที่จะ compile แต่ import ของ Java เป็นการบอก compiler ให้ทราบว่าเวลา compile ให้เชื่อมโยงคลาสต่าง ๆ จากภายนอก (packages) มา compile ด้วยเท่านั้น

สำหรับในส่วนของโครงสร้างโปรแกรมแล้ว Java จะมีความแตกต่างกับ C++ อยู่มากทีเดียว อาทิเช่น

- Java ไม่มีความสามารถทำ Multi-Inheritance
- ทุกอย่างใน Java เป็นคลาสหมด ไม่มีฟังก์ชันสาธารณะ (global function) ไม่มีข้อมูลสาธารณะ (global data) ถ้าต้องการทำให้เป็น global ต้องใช้คำสั่ง static แทน
- Java ไม่มีคำสั่ง struct, enum, unions, inline, goto เหมือนกับ C++
- การกำหนดคลาส (class definitions) ใน Java ไม่มีเครื่องหมาย ; ที่ท้ายของคำสั่ง
- Java ไม่มีการประกาศคลาส (class declarations) เหมือนใน C++
- ไม่มีตัวดำเนินการแยกขอบเขต (::) ใน Java (scope resolution operator)
- Java ไม่สนับสนุน default arguments
- ฯลฯ

แต่สิ่งที่สำคัญและควรจะจดจำคือ คลาสที่เป็นแบบ public นั้นจะใส่ได้เพียงที่เดียวเท่านั้นใน 1 file และคลาสนั้นจะต้องมีชื่อเหมือน



รูปที่ 3 แสดงตัวอย่างการอ้างชื่อ package, ชื่อคลาสและ method ของ Java

กับชื่อของ file ด้วย (ต้องเหมือนกันทุกประการ คือถ้าเขียนชื่อ file ด้วยตัวพิมพ์เล็ก ชื่อคลาสก็จะต้องเป็นตัวพิมพ์เล็กด้วย) ในกรณีจะสร้าง java applet จะต้อง import คลาส java.applet และในคลาส public ต้อง extends จากคลาส applet ด้วย

Java Packages and Libraries

นอกจากโครงสร้างของภาษาและความสามารถที่มีอยู่ในตัวภาษา Java แล้ว ยังมีส่วนที่สำคัญมากในการพัฒนาโปรแกรมคือ ห้องสมุดคลาส (Class Libraries) ห้องสมุดนี้จะประกอบด้วยคลาส (Classes) ต่าง ๆ แล้วแต่จุดประสงค์ของไลบรารีนั้น ๆ

ใน Java มีห้องสมุดคลาสมาตรฐาน หรือเรียกอีกอย่างหนึ่งว่า Java API (Application Programming Interface) ประกอบด้วยคลาสต่าง ๆ ที่จำเป็นและสนับสนุนการพัฒนาโปรแกรมโดยใช้ Java และแต่ละคลาสที่อยู่ใน Java API นี้จะถูกแยกเก็บเป็น package เช่น คลาส String ถูกเก็บไว้อยู่ใน package ชื่อ java.lang และคลาสที่อยู่ใน package เดียวกันจะเป็นคลาสที่เกี่ยวข้องกันในเรื่องเดียวกัน บริษัท Sun Microsystems ผู้คิดค้นภาษา Java เป็นผู้กำหนด Java API ขึ้นมาและนำรวมไว้ในชุดพัฒนาโปรแกรมภาษา Java ซึ่งถูกเรียกว่า Java Development Kit (JDK) และแจกจ่ายให้กับผู้สนใจในการเขียนโปรแกรมภาษา Java ได้นำไปใช้พัฒนาโปรแกรมภาษา Java โดยไม่คิดค่าใช้จ่าย รายละเอียดเพิ่มเติมพบได้ที่เว็บไซท์ java.sun.com

Packages

Package ในภาษา Java คือกลุ่มของคลาสหลาย ๆ คลาสรวมกันอยู่ ซึ่งคลาสที่อยู่ใน package หนึ่ง ๆ นั้น อาจจะเกี่ยวข้องกันแบบ Inheritance หรือไม่ก็ได้ โปรแกรมเมอร์สามารถกำหนด package และเพิ่มคลาสเข้าไปใน package ได้เอง

ไฟล์โปรแกรมภาษา Java จะถูกเก็บไว้ในไฟล์ที่มีนามสกุลเป็น .java ซึ่งในไฟล์นั้นอาจจะมีคลาสหรืออินเทอร์เฟสมากกว่า 1 ตัวก็ได้ แต่สามารถถูกกำหนดเป็นแบบ public ได้เพียงคลาสหรือว่าอินเทอร์เฟสเดียวเท่านั้น นอกจากนั้นแล้วชื่อไฟล์จะต้องมีชื่อเหมือนชื่อคลาสหรืออินเทอร์เฟสที่ public รวมกับนามสกุล .java เมื่อแปล (compile) ไฟล์โปรแกรมภาษา Java แล้วจะได้ Java byte code เก็บอยู่ในไฟล์ที่มีชื่อเหมือนเดิมแต่มีนามสกุลเป็น .class ไฟล์ที่ได้นี้เรียกอีกอย่างหนึ่งว่าคลาสไฟล์ (Class File) คลาสไฟล์ที่ได้นี้จะถูกเก็บในไดเรคทอรี (Directory) เหมือนกับชื่อ package เช่น มีการอ้างถึงคลาส david.games.tetris.SoundEffects คลาสไฟล์ของคลาสนี้จะต้องเก็บไว้ที่ไดเรคทรอรี david/games/tetris หรืออ้างอิงจาก CLASSPATH environment variable และตัวอย่างของการอ้างชื่อ package, ชื่อคลาสและ method ของ Java แสดงในรูปที่ 3

การกำหนด package สามารถทำได้โดยใช้ package statement เพื่อบอกว่าคลาสทั้งหมดในไฟล์นั้นเป็นของ package ใด ๆ ดังนี้

```
package package-name;
```

ซึ่ง package statement นี้ต้องกำหนดไว้ที่ต้นไฟล์และต้องมีเพียง 1 package statement เท่านั้นต่อหนึ่งไฟล์

ไฟล์ภาษา Java ใด ๆ ที่ถูกกำหนดไว้ให้อยู่ใน package เดียวกันสามารถเข้าถึง (Access) คลาสหรืออินเทอร์เฟสทั้งหมดที่อยู่ใน package นั้นแม้ว่าจะเก็บอยู่คนละไฟล์

การอ้างถึงคลาสที่อยู่ใน package

การอ้างถึงคลาสที่อยู่ใน package สามารถทำได้ดังนี้

- การอ้างแบบเต็ม

```
java.lang.System.out.printfn();
```

- การอ้างโดยใช้ import statement

```
- import package.class;
  import java.util.Random;
- import package.*;
  import java.applet.*;
```

หมายเหตุ :- package java.lang จะถูก import เข้าไปในโปรแกรมที่เขียนโดยอัตโนมัติ

Java API packages

Java 1.1 API ประกอบด้วยคลาส และอินเทอร์เฟสสำหรับใช้ในการพัฒนาโปรแกรมทั้งสิ้น 23 packages ซึ่งมีรายละเอียดดังนี้

1. java.applet – ประกอบด้วยคลาส Applet ซึ่งเป็น superclass ของทุก ๆ applet และ Interface 3 ตัวที่เกี่ยวข้องกับ package นี้สนับสนุนการเขียน applet ซึ่งคลาส Applet นี้จะกำหนดอินเทอร์เฟส ระหว่าง applet กับ environment ของ applet นั้น

2. java.awt – Abstract Window Toolkit เป็น package ที่ใช้

สำหรับการเขียนโปรแกรมที่มี GUI (Graphical User Interface) พื้นฐาน อย่างเช่น หน้าต่าง ปุ่ม เป็นต้น

3. java.awt.datatransfer – คลาสที่อยู่ใน package นี้จะกำหนดโครงสร้างพื้นฐาน (Generic Framework) สำหรับการถ่ายเทข้อมูลของโปรแกรม ซึ่งคลาสจะสนับสนุน clipboard-based cut-and-paste model หรือผ่านวิธี drag-and-drop

4. java.awt.event – คลาสและอินเทอร์เฟสในคลาสนี้ เกี่ยวกับการจัดการกับ Event

5. java.awt.image – ประกอบด้วยคลาสที่สนับสนุนการจัดการกับ Image เช่น Image Scaling

6. java.awt.peer – เป็นคลาสเกี่ยวกับ GUI Interfaces สำหรับ platform independence เช่น Popup Menu, การ scrolling

7. java.beans – ประกอบด้วยคลาสและอินเทอร์เฟสซึ่งทำตามข้อกำหนดของ Java Bean API เช่น การสร้าง และการใช้ embeddable, reusable software components เป็นต้น

8. java.io – ประกอบด้วยคลาสเกี่ยวกับ input และ output แบบต่าง ๆ

9. java.lang – ประกอบด้วย base types และคลาสที่เป็นพื้นฐานของภาษา Java และ package นี้จะถูก import ในโปรแกรมเสมอโดยอัตโนมัติ

10. java.lang.reflect – ประกอบด้วยคลาสที่แทน fields, methods และ constructors ของคลาส ส่งผลให้โปรแกรมสามารถหาข้อมูลเกี่ยวกับ object, array, method, constructor หรือ field ใด ๆ

11. java.math – ประกอบด้วยคลาสทางคณิตศาสตร์ของตัวเลขจำนวนเต็มแบบกำหนดขนาดได้ และตัวเลขทศนิยมแบบกำหนดความละเอียดได้

12. java.net – จัดเตรียมโครงสร้างพื้นฐานที่ยืดหยุ่นและมีประโยชน์อย่างมากเกี่ยวกับ Networking เช่น Socket

13. java.rmi – ประกอบด้วยคลาสที่สนับสนุนการใช้งาน remote objects และ Remote Method Invocation (RMI) ในการสื่อสารระหว่างเครื่องคอมพิวเตอร์ที่ Java ทำงาน

14. java.rmi.dgc – คลาสและอินเทอร์เฟสใน package นี้เกี่ยวข้องกับ Distributed Garbage Collection (DGC) ซึ่งมีความเกี่ยวข้องกับ RMI Classes

15. java.rmi.registry – ประกอบด้วยคลาสและอินเทอร์เฟสสำหรับ Java Client ในการค้นหา Remote Object โดยชื่อ หรือสำหรับ Java Server ใช้ทำ advertise services ของตัวเอง

16. java.rmi.server – ประกอบด้วยคลาสที่เป็นหัวใจของ Remote Method Invocation ให้โปรแกรมสามารถสร้าง object ที่สามารถถูกเรียกใช้งานโดยอีกโปรแกรมโดยผ่านทางระบบเครือข่าย

17. java.security – ประกอบด้วยคลาสและอินเทอร์เฟสที่เกี่ยวกับ cryptographic security คือ public and private keys, certificates, message digests และ digital signatures

18. java.security.acl – ประกอบด้วยอินเทอร์เฟส และ exceptions สำหรับจัดการกับ access control lists

19. java.security.interfaces – ประกอบด้วยอินเทอร์เฟส ที่จำเป็นสำหรับ Java Security API's Implementation-independent design

20. java.sql – เป็น API เกี่ยวกับ Java Database Connectivity (JDBC) ซึ่งจะมีคลาสและอินเทอร์เฟสที่ทำให้โปรแกรมสามารถส่ง SQL Queries ไปยัง Database และรับผลลัพธ์ของการ Query นั้น

21. java.text – ประกอบด้วยคลาสและอินเทอร์เฟสที่มีประโยชน์สำหรับการเขียนโปรแกรมแบบที่ใช้ได้ทุกภาษา (Internationalized Programs) ที่จัดการโลคอล (Local) เอง อย่างเช่น การจัดรูปแบบของวันที่และเวลา และตัวอักษร เป็นต้น

22. java.util – ประกอบด้วยคลาส Utility ต่าง ๆ เช่น Date, BitSet เป็นต้น และ Collection Classes เช่น Dictionary และ Vector เป็นต้น

23. java.util.zip – ประกอบด้วยคลาสสำหรับการคำนวณหา Checksums ของ Streams ของข้อมูล, Compressing/Decompressing และ Archiving/Unarchiving โดยใช้ ZLIB compression library และใช้ ZIP และ GZIP file formats

ปัจจุบัน Java เป็นภาษาที่ได้รับความนิยมมากภาษาหนึ่ง ด้วยเหตุผลหลักก็คือสามารถพัฒนาครั้งเดียว และนำไปใช้ได้ในหลาย ๆ แพลตฟอร์ม ใช้แนวคิดทางโปรแกรมเชิงวัตถุทำให้เข้าใจง่าย และยังสามารถนำสิ่งที่สร้างไว้มาใช้ใหม่ หรือเพิ่มเติมความสามารถลงไปได้ รวมทั้งยังรองรับการทำงานในแบบต่าง ๆ ได้ เช่น การโต้ตอบแบบมัลติมีเดีย ทำโปรแกรมเพื่อการศึกษา การพัฒนาเว็บเพจ พัฒนาเกมและความบันเทิง เป็นต้น และด้วยความสามารถที่ทำงานได้โดยไม่ขึ้นอยู่กับลักษณะของแพลตฟอร์ม Java จึงถูกใช้เป็นโปรแกรมจัดการอุปกรณ์ และฝังตัวเข้าในระบบสำหรับผลิตภัณฑ์อิเล็กทรอนิกส์ต่าง ๆ เช่น อุปกรณ์ Hand-held โทรศัพท์ และ VCR ในอนาคตความหวังของ Sun ที่จะให้ Java อยู่ในทุกหนทุกแห่ง (Java everywhere) ก็อาจจะเป็นจริงขึ้นมาก็ได้ ซึ่งอาจจะเรียกได้ว่าเป็นภาษาคอมพิวเตอร์สำหรับอนาคตโดยแท้

หนังสืออ้างอิง

Core JAVA second edition โดย Gray Cornell,
Cay S. Horstmann Thinking in Java โดย Bruce Eckel
http://www.javasoft.com
http://java.sun.com